An investigation into trapezoid packing

By **Rhyd Lewis**, X. Song,
K. Dowsland, J. Thompson

Operational Research Group
Cardiff School of Mathematics,
Cardiff University

Gower Optimal Algorithms Ltd.

Grouping Problems

- Partition a set of items I into an exhaustive set of mutually exclusive subsets (groups) $\boldsymbol{U} = \{U_1, \ldots, U_{|\boldsymbol{U}|}\}$ such that $|\boldsymbol{U}|$ is minimised and

$$\forall U_i \in \boldsymbol{U}, \quad U_i \in \boldsymbol{F}$$

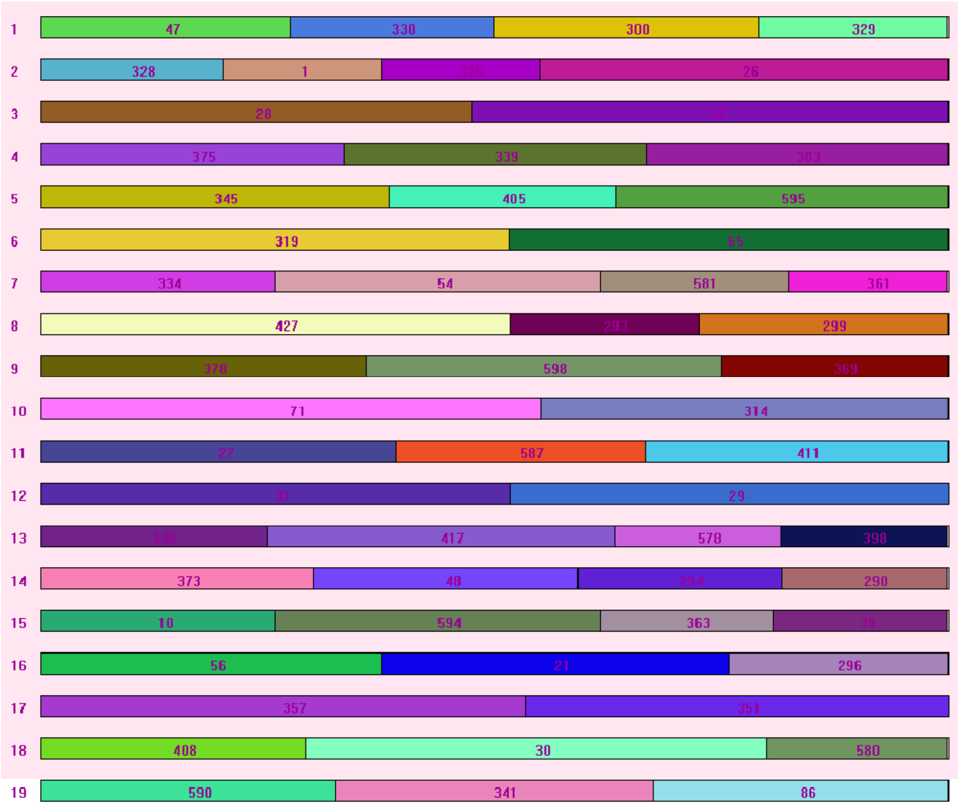
  where $\boldsymbol{F}$ is the set of all feasible groups.

- For example, in one dimensional bin packing and stock cutting, given a "size" s_j for each item $j \in I$,

$$U_i \in \boldsymbol{F} \Leftrightarrow \sum_{j \in U_i} s_j \leq C$$

  for some predefined constant C

Grouping Problems



Example 1-D Bin Packing Solution (most items are different)

- Partition a set of items I into an exhaustive set of mutually exclusive subsets (groups) $\boldsymbol{U} = \{U_1, \ldots, U_{|\boldsymbol{U}|}\}$ such that $|\boldsymbol{U}|$ is minimised and

$$\forall U_i \in \boldsymbol{U}, \quad U_i \in \boldsymbol{F}$$

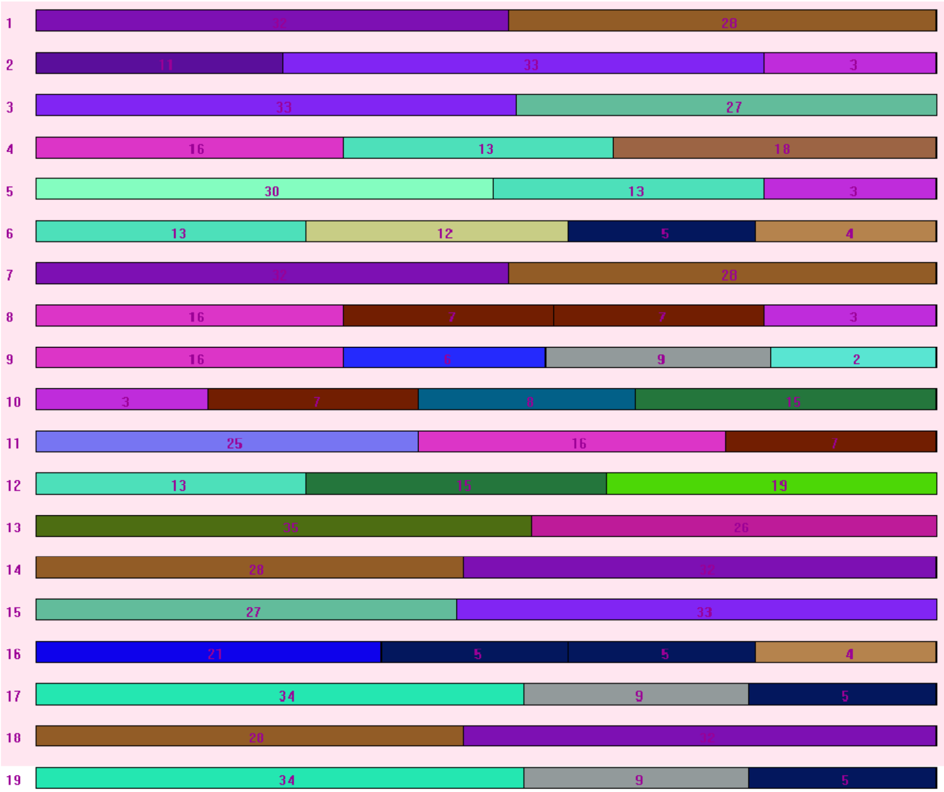
where $\boldsymbol{F}$ is the set of all feasible groups.

- For example, in one dimensional bin packing and stock cutting, given a “size” s_j for each item,

$$U_i \in \boldsymbol{F} \Leftrightarrow \sum_{j \in U_i} s_j \leq C$$

for some constant C

Grouping Problems



<u>*Example 1-D Stock Cutting Solution (many items of same dimensions)*</u>

- Partition a set of items I into an exhaustive set of mutually exclusive subsets (groups) $\boldsymbol{U} = \{U_1, \ldots, U_{|\boldsymbol{U}|}\}$ such that $|\boldsymbol{U}|$ is minimised and

$$\forall U_i \in \boldsymbol{U}, \quad U_i \in \boldsymbol{F}$$

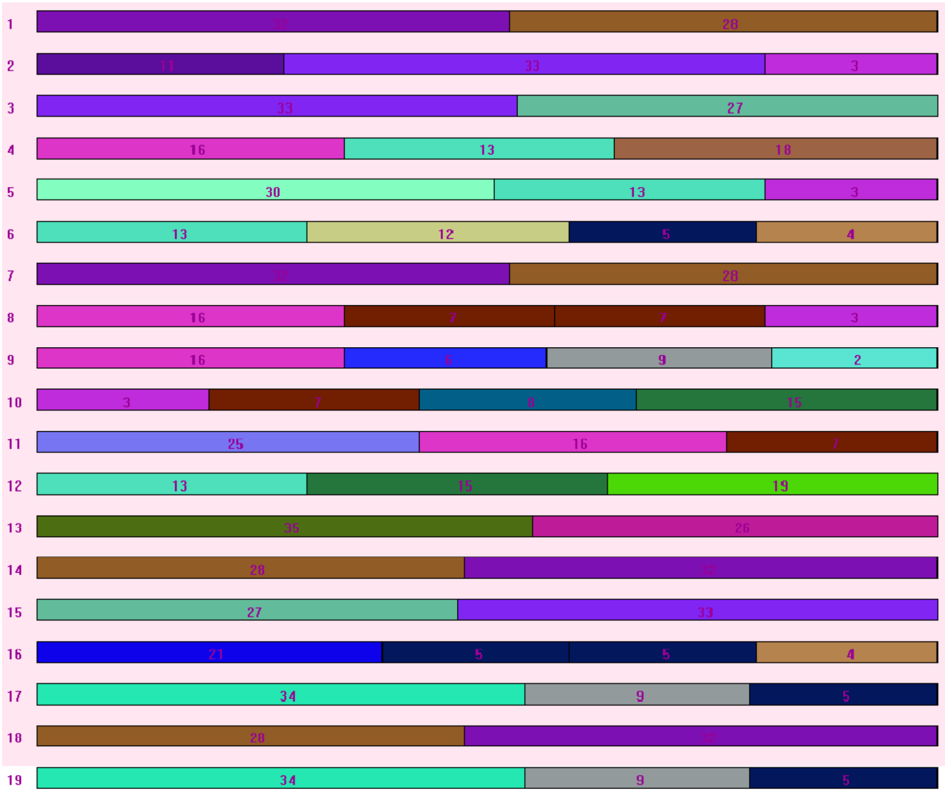
where $\boldsymbol{F}$ is the set of all feasible groups.

- For example, in one dimensional bin packing and stock cutting, given a “size” s_j for each item,

$$U_i \in \boldsymbol{F} \Leftrightarrow \sum_{j \in U_i} s_j \le C$$

for some constant C

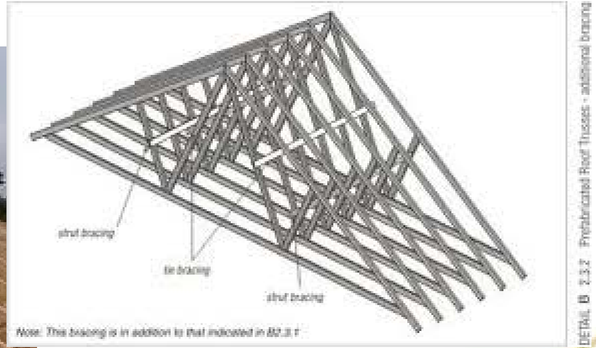
Grouping Problems

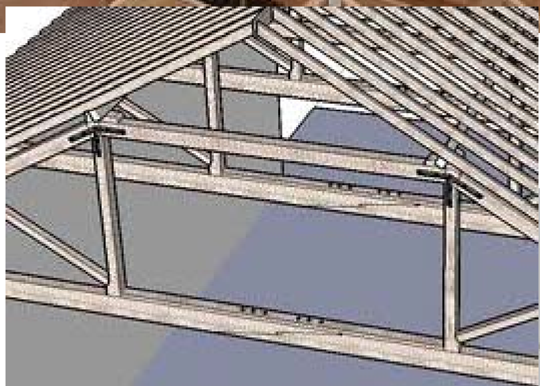


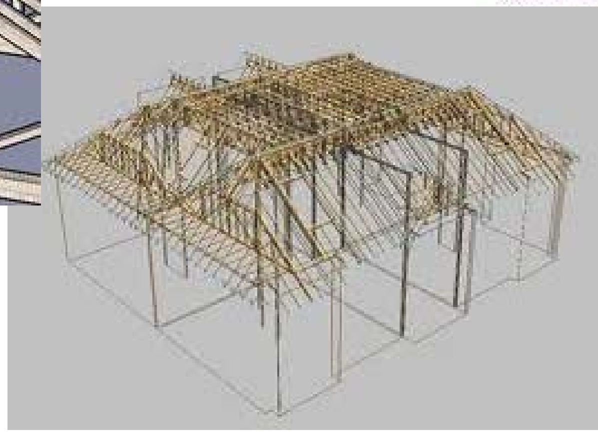
Example 1-D Stock Cutting Solution (many items of same dimensions)

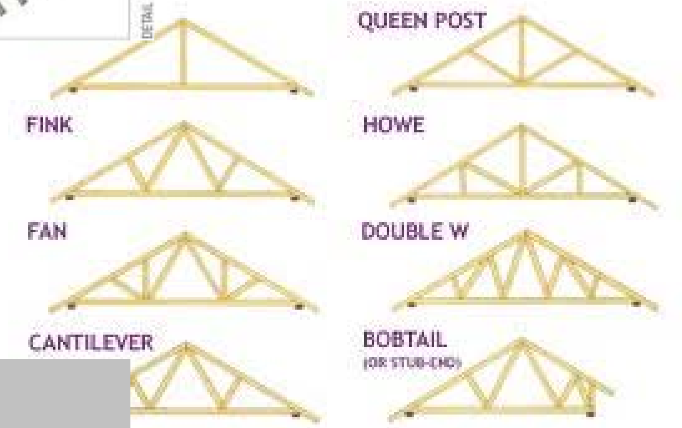
- In these cases group ordering unimportant.
- Also, the order and orientation of items *within* each group irrelevant
- But what if it is?

Truss Cutting Problem

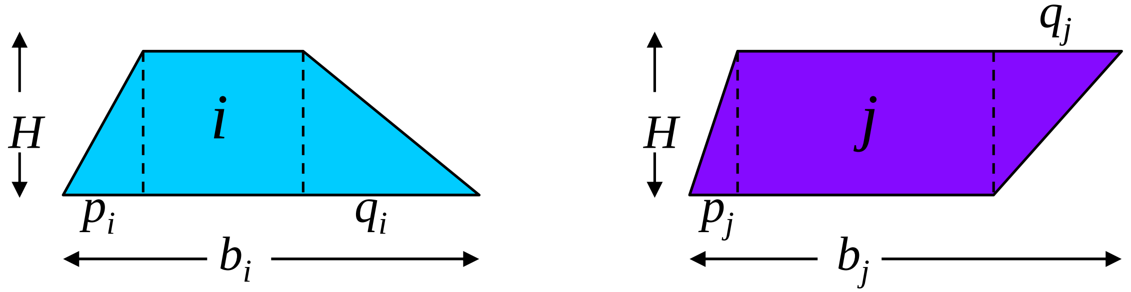




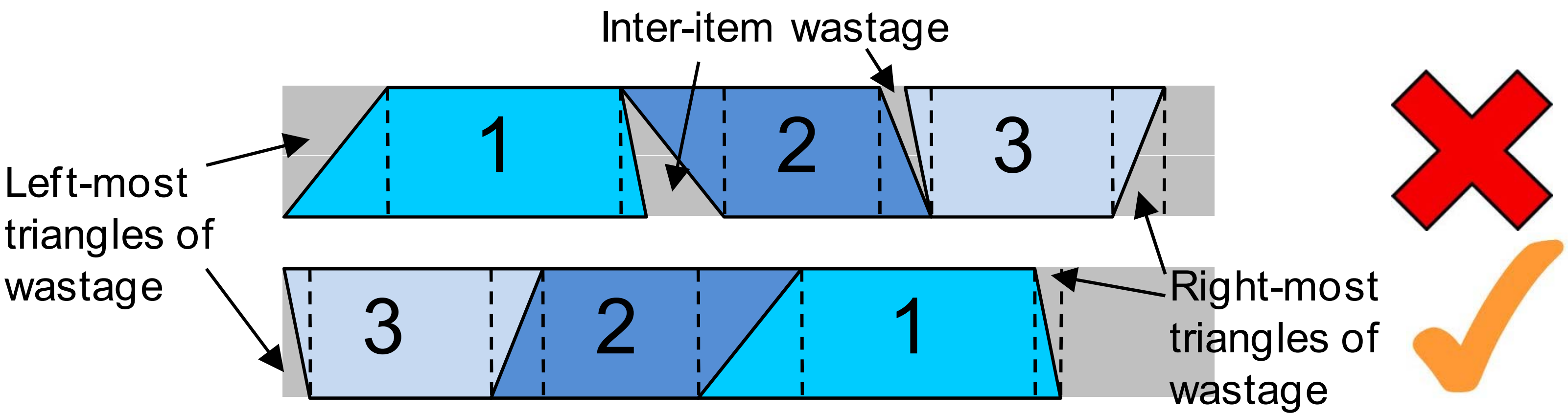




Truss Cutting Problem

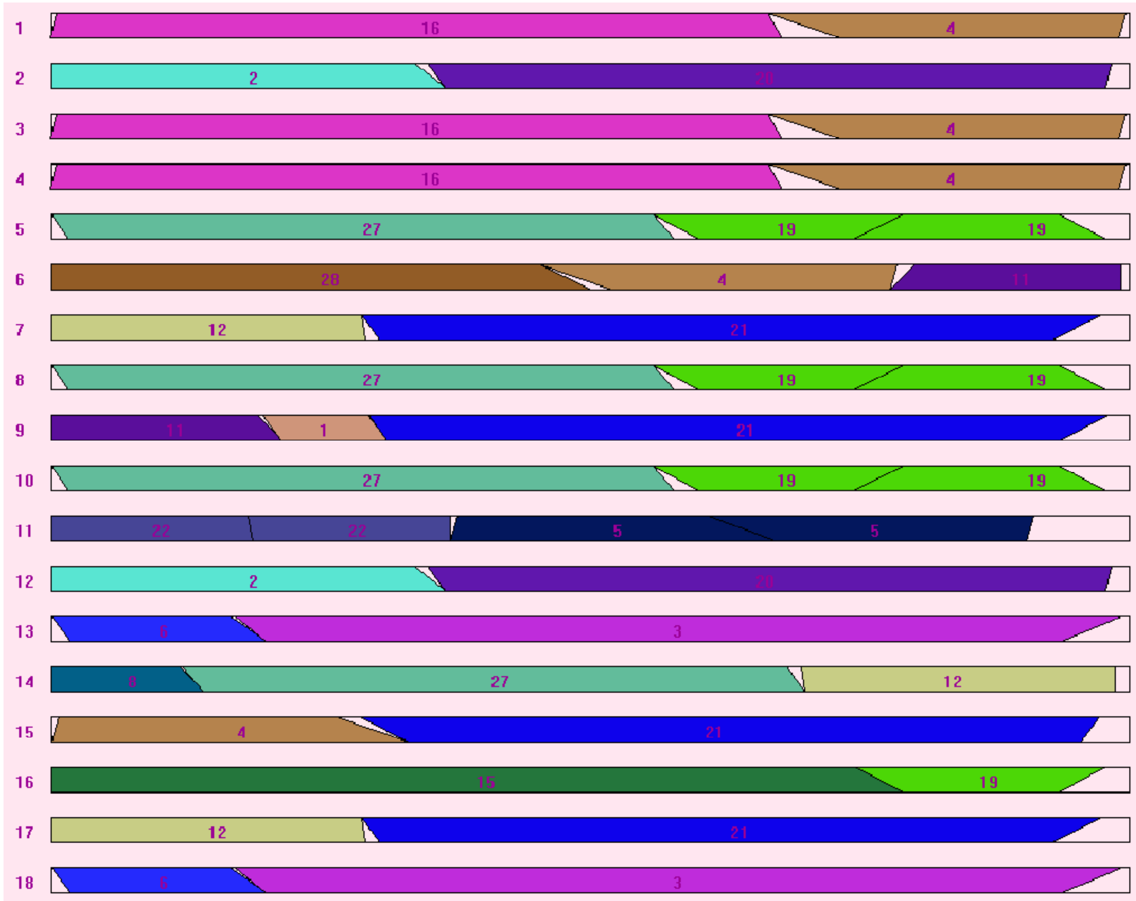


- Given a set of n trapezia of height H, assign each to one of a set of $(W \times H)$ stocks such that the number of stocks is minimised
- Order and orientation of items could be **very important** here…

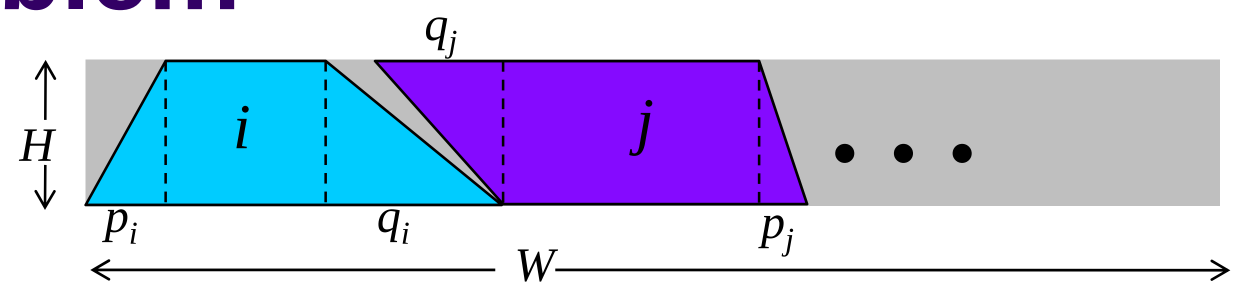


Example Solution…

- There are two issues:
  - Which items should be grouped together? (the **grouping problem)**
  - Given a group of items, how should these be arranged on a stock? (The **truss sub-problem**)



Understanding the Truss Sub-problem



- Items have 4 orientations, but we only need to consider 2
- Inter-item wastage in the above is simply $|q_i - q_j|$
- Wastage w = all inter-item wastage plus the LHS and RHS
- **Problem Def:** Given a group of items U_i such that:

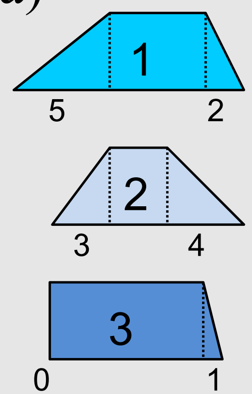
$$A(U_i) = \sum_{j \in U_i} A(j) \leq HW$$

is $U_i \in \boldsymbol{F}$? That is, can we determine an item arrangement such that:

$$A(U_i) + w \leq HW$$

Understanding the Truss Sub-problem

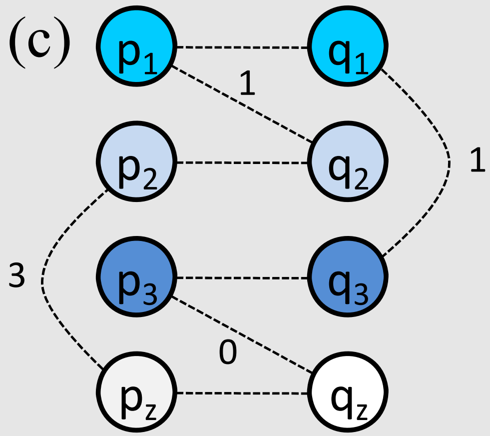
(a)



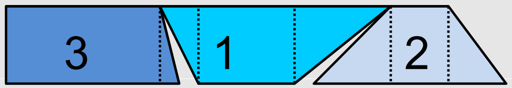
(b)

	p_1	q_1	p_2	q_2	p_3	q_3	p_z	q_z
p_1	∞	$-\infty$	2	1	5	4	5	5
q_1	$-\infty$	∞	1	2	2	1	2	2
p_2	2	1	∞	$-\infty$	3	2	3	3
q_2	1	2	$-\infty$	∞	4	3	4	4
p_3	5	2	3	4	∞	$-\infty$	0	0
q_3	4	1	2	3	$-\infty$	∞	1	1
p_z	5	2	3	4	0	1	∞	$-\infty$
q_z	5	2	3	4	0	1	$-\infty$	∞

(c)



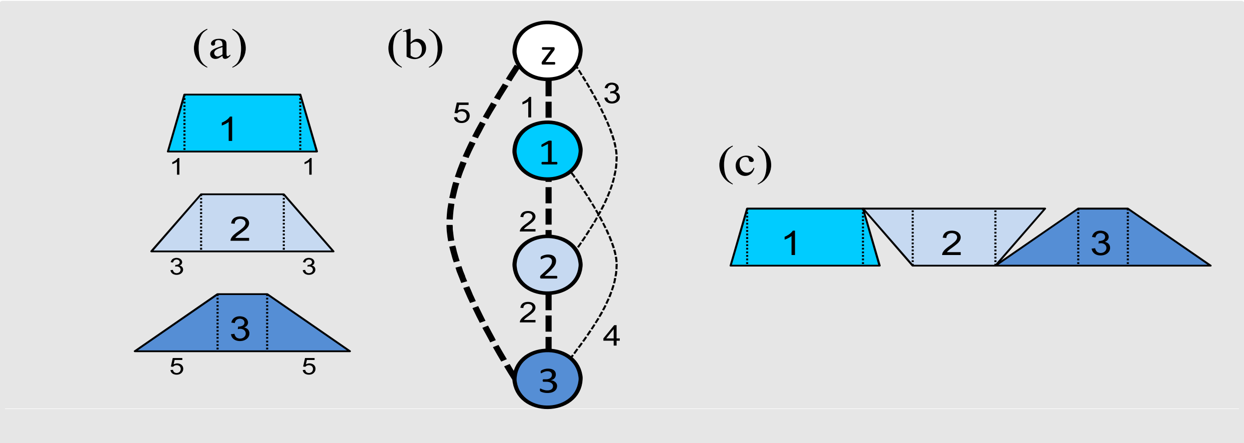
(d)



- The Truss sub-problem is a type of TSP
  - Each projection is a "city" ; arcs between cities correspond to wastage; arcs on same shape are set to $-\infty$
  - Two dummy cities (p_z, q_z) represent wastage on the LHS and RHS
  - Can a "valid" route of less than w be achieved (discounting $-\infty$ arcs)?
  - There are $(2^{|U_i|}|U_i|!)/2$ possible valid routes in total (24 in e.g. above)

Special Cases of the Truss Sub-Problem

- If all projections are of size zero, all items will be rectangular and thus all arrangements are optimal (trivial)
- If all shapes are symmetrical (Isosceles trapezia and parallelograms), optimal arrangements can be found by placing items onto the stock in projection-size order…



- However, the general sub-problem is more tricky. Should it be addressed using heuristics, or should we try to use exact methods?

Experimental Set-up

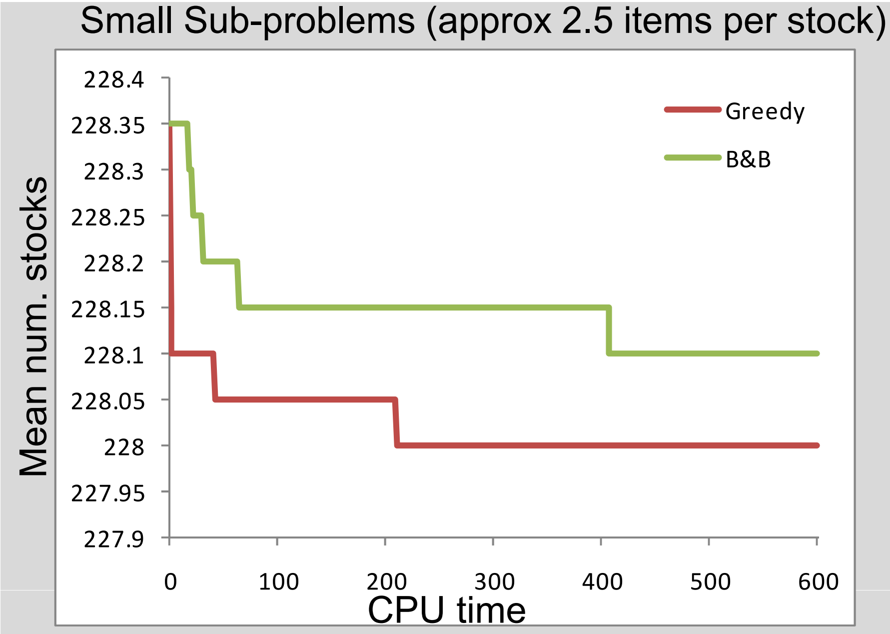
- Two Algorithms were implemented:
  - **A)** Attempt to "solve" sub-problems using simple greedy heuristics. If no suitable arrangement is found, then assume $U_i \notin \boldsymbol{F}$
  - **B)** As above but also employ branch-and-bound for increased accuracy (***Xpress***). During a run, save information on previous B&B applications to avoid duplication of effort.
- Both embedded in a local search algorithm originally designed for 1D bin packing (Lewis, R. (2009), ***Comp. and Op. Res.***, vol. 36(7), pp 2295-2310).
- Over 1000 problem instances were generated attempting to simulate the "real" problem (not totally satisfactory…)

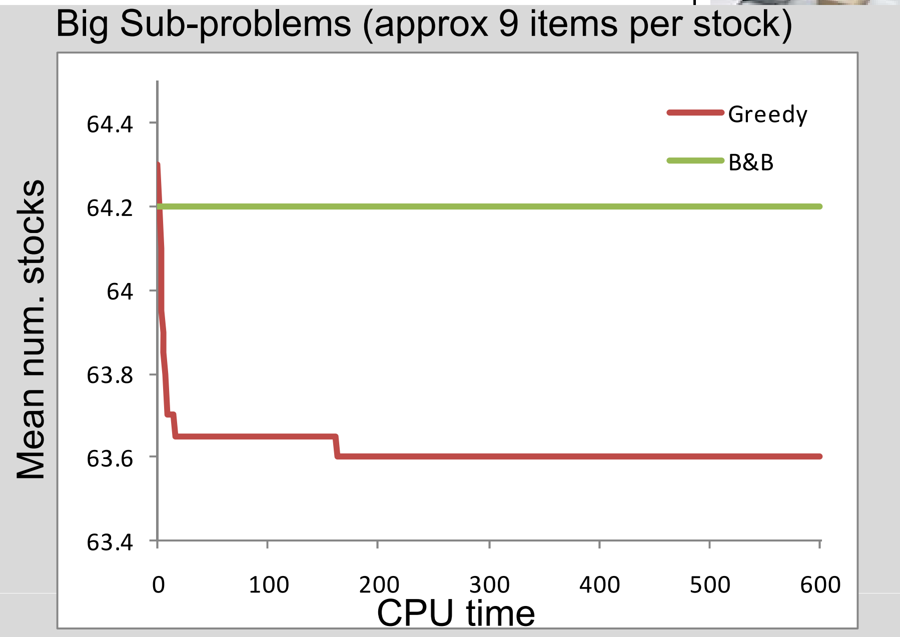
Example Solutions

- Sub-problems in the simulated “real” problems typically very small…
  - With $n = 500$, theoretical minimum never reached, but answers are within other generated upper and lower bounds

- We also produced further problems with larger sub-problems…
  - With $n = 500$, theoretical minimum reached in over half of cases

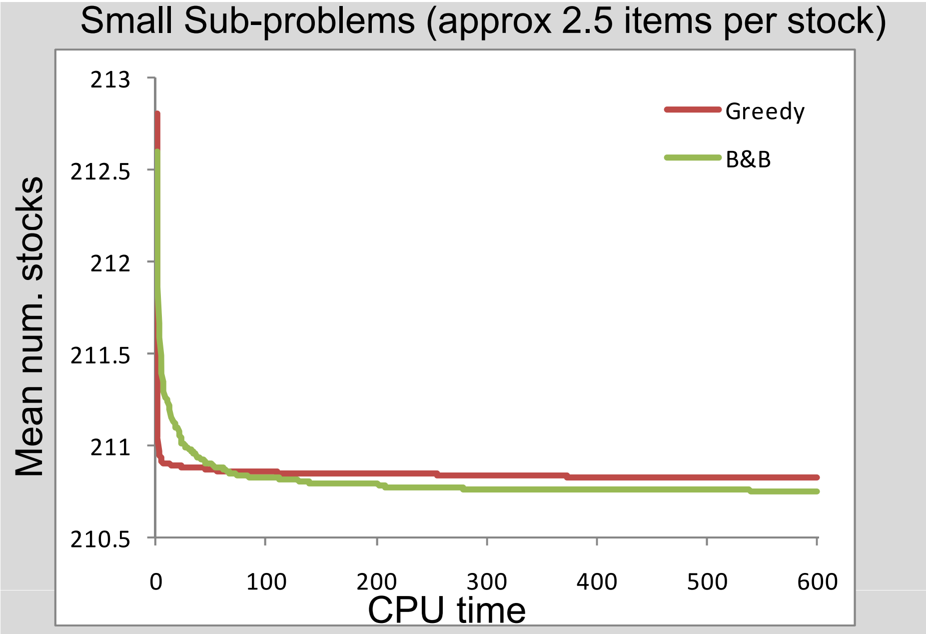
Run Profiles (no repeated items)

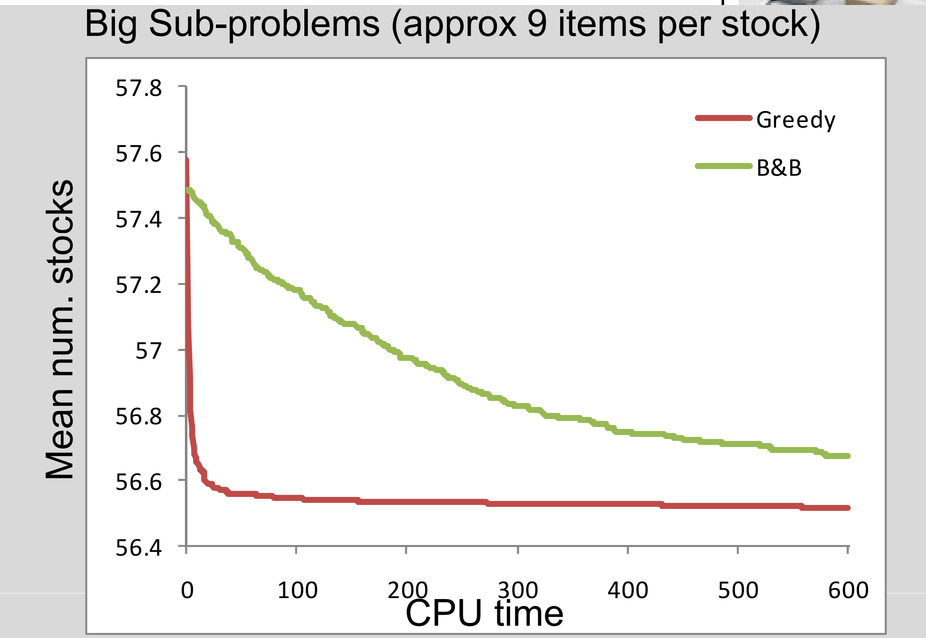




- Algorithm using branch and bound is much slower
- This is accentuated in problems with smaller trusses (larger sub-problems)
- Despite its increased accuracy, the cost of B&B means less iterations of the local search algorithm take place over time

Run Profiles (repeated items)

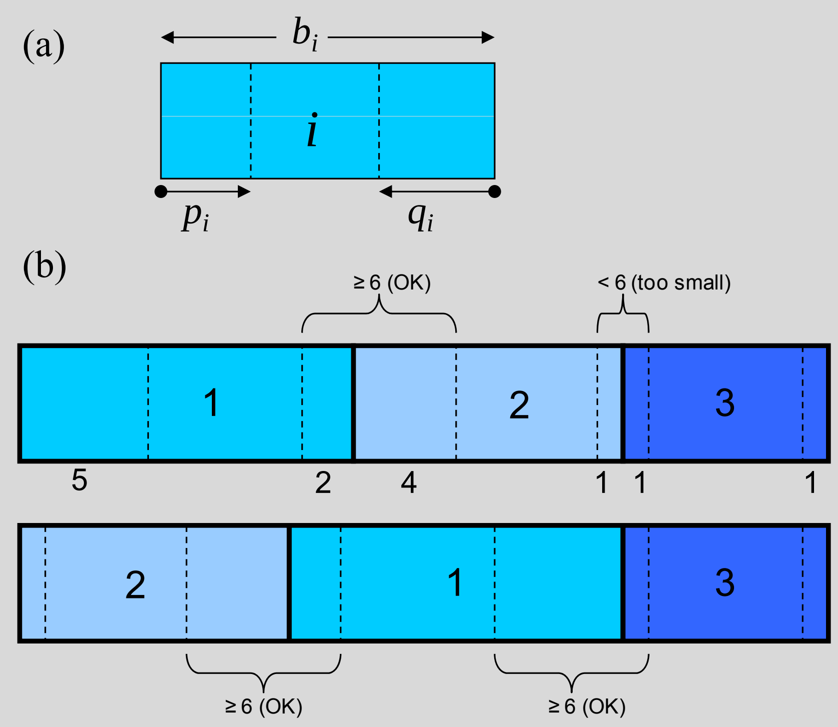




- However, performance of B&B algorithm improves when items are repeated in the original problem (the same sub-problems are encountered more often)
- Though larger sub-problems still lead to degradation of performance

A Related Problem (Goulimis (2004) “Minimum Score Separation – an open combinatorial optimisation problem” *J. of Op. Res. Soc*, vol. 55, pp 1367–1368)

- In the manufacture of cardboard boxes, we need to cut rectangular strips from stocks which are also simultaneously scored.
- Each item has two “score distances”. However, scores cannot be too close together when cut
- Thus, like the truss problem, item order and orientation is considered when deciding whether $U_i \in \boldsymbol{F}$
- Our truss algorithm is applicable here with minor alterations



Example formation of 3 items.
Here, scores need to be greater or equal to 6

Conclusions

- These problems are similar to bin packing and stock cutting problems. But they also involve potentially complex sub-problems
- During optimisation, many sub-problems are faced. Thus an appropriate balance needs to be struck between accuracy and expense in solving these.
- Saving information on the outcome of previous sub-problems has benefits when items are repeated in a problem
- Future Issues:
  - Truss Problem may have multiple sized stocks in practice.
  - Might also be interested in using "left-overs" for solving the next problem
  - Can improved performance be achieved? In particular by improving the methods by which sub-problems are solved.

Conclusions

- These ... d stock cutting ... ially comple...
- During ... faced. Thus an appr... tween accura...
- Saving ... s sub-problen... ed in a problen...

Thank you…

- Future Issues:
  - Truss Problem may have multiple sized stocks in practice.
  - Might also be interested in using “left-overs” for solving the next problem
  - Can improved performance be achieved? In particular by improving the methods by which sub-problems are solved.